Claims

We claim:

1. A method for improving routing operations in dynamic routing tables comprising:

(a) establishing s to be used in a prefix partitioning scheme;

(b) partitioning at least one prefix into up to min(n, $2^s + 1$) partitions;

(c) assessing prefix length with regard to s;

(d) storing the prefix whose length is larger than or equal to s in a partition that corresponds to the value of the prefix's first s bit;

(e) storing prefixes whose length is smaller than s in a first designated partition; and

(f) representing the prefixes in each partition using a priority search tree, wherein operation on the priority search tree constitutes defining a rectangle comprising the prefixes associated with a desired destination address.

2. The method according to claim 1, wherein the priority search tree is a red-black priority-search tree (RBPST) that is performed according to the following formula: *minXinRectangle*(x_{left}, x_{right}, y_{top}), wherein x_{left}, x_{right}, y_{top} are points that lie in a rectangle defined by $x_{left} = d$, $x_{right} = \infty$, $y_{top} = d$, said points are provided by performing the following formula: *enumerateRectangle*($2^W d - d + 2^W - 1$, ∞, d), wherein d is the destination and W is the number of bits in the destination.

3. The method according to claim 1, wherein the priority search tree is a radix priority-search tree (RPST).

4. The method according to claim 1, further comprising the steps of inserting a new prefix into the priority search tree, wherein every point set of the new prefix set is unique, and representing the new prefix set as the priority search tree.

5. The method according to claim 1, further comprising the step of deleting a prefix from the priority search tree.

6. A method for improving routing operations in dynamic routing tables comprising:

(a) representing at least one prefix as a range with end points;

(b) establishing at least one interval using the end points; and

(c) partitioning the intervals into interval-based dynamic router table data structures.

7. The method according to claim 6, wherein the step of partitioning the intervals into interval-based dynamic router table data structures comprises:

(a) storing the intervals in a node of a basic interval tree;

(b) storing the prefixes whose ranges are represented by the intervals in a collection of prefix trees;

(c) establishing a header node and a prefix node for each prefix tree;

(d) establishing a pointer to the prefix tree for an interval; and

(e) storing the pointer with the interval in the basic interval tree node.

8. The method according to claim 7, further comprising the step of establishing a pointer to *W*-length prefixes that correspond to the end points of intervals whose length is *W*.

9. The method according to claim 7, further comprising the step of finding the longest matching prefix, wherein the basic interval tree is searched for the interval and header to the appropriate prefix tree and the longest matching prefix.

10. The method according to claim 7, further comprising the step of inserting a new prefix, wherein the intervals that are created by the new prefix are stored in the basic interval tree; wherein the new prefix is stored in a prefix tree; wherein the pointer to the prefix tree for the new intervals is established and stored with the new intervals in the basic interval tree.

11. The method according to claim 7, further comprising the step of deleting a prefix, wherein the intervals in the basic interval tree that correspond to the prefix to be deleted are deleted; wherein the prefix is deleted from the prefix tree; wherein the pointer to the prefix tree for the deleted intervals are deleted from the basic interval tree node.

12. A computer program product recorded on computer readable medium for routing packets comprising; a computer readable medium for receiving packets specifying prefixes and decision rules for the prefixes; a computer readable medium for establishing *s* bits to be used in a prefix partitioning scheme; a computer readable medium for matching, inserting, or deleting prefixes in a partitioning tree; and a computer readable medium for performing steps of multilevel partitioning, said steps comprising:

(a) establishing *s* to be used in a prefix partitioning scheme;

(b) partitioning at least one prefix into up to min(n, $2^s + 1$) partitions;

(c) assessing prefix length with regard to *s*;

(d) storing the prefix whose length is larger than or equal to *s* in a partition that corresponds to the value of the prefix's first *s* bit;

(e) storing prefixes whose length is smaller than *s* in a first designated partition; and

(f) representing the prefixes in each partition using a priority search tree, wherein operation on the priority search tree constitutes defining a rectangle comprising the prefixes associated with a desired destination address.

13. The computer program according to claim 12, wherein the priority search tree is a red-black priority-search tree (RBPST) that is performed according to the following formula: $minXinRectangle(x_{left}, x_{right}, y_{top})$, wherein $x_{left}, x_{right}, y_{top}$ are points that lie in a rectangle defined by $x_{left} = d$, $x_{right} = \infty$, $y_{top} = d$, said points are provided by performing the following formula: *enumerateRectangle*($2^W d - d + 2^W - 1, \infty, d$), wherein *d* is the destination and W is the number of bits in the destination.

14. The computer program according to claim 12, wherein the priority search tree is a radix priority-search tree (RPST).

15. The computer program according to claim 12, wherein the computer readable medium further performs the steps of inserting a new prefix into the priority search tree, wherein every point set of the new prefix set is unique, and representing the new prefix set as the priority search tree.

16. The computer program according to claim 12, wherein the computer readable medium further performs the steps of deleting a prefix from the priority search tree.

17. A computer program product recorded on computer readable medium for routing packets comprising; a computer readable medium for receiving packets specifying prefixes and decision rules for the prefixes; a computer readable medium for establishing *s* bits to be used in a prefix partitioning scheme; a computer readable medium for matching, inserting, or deleting prefixes in a partitioning tree; and a computer readable medium for performing steps of multilevel partitioning, said steps comprising:

(a) representing at least one prefix as a range with end points;

(b) establishing at least one interval using the end points; and

(c) partitioning the intervals into interval-based dynamic router table data structures.

18. The computer program according to claim 17, wherein the computer readable medium further performs the steps of:

(a) storing the intervals in a node of a basic interval tree;

(b) storing the prefixes whose ranges are represented by the intervals in a collection of prefix trees;

(c) establishing a header node and a prefix node for each prefix tree;

(d) establishing a pointer to the prefix tree for an interval; and

(e) storing the pointer with the interval in the basic interval tree node.

19. The computer program according to claim 17, wherein the computer readable medium further performs the steps of establishing a pointer to *W*-length prefixes that correspond to the end points of intervals whose length is *W*.

20. The computer program according to claim 17, wherein the computer readable medium further performs the steps of finding the longest matching prefix, wherein the basic interval tree is searched for the interval and header to the appropriate prefix tree and the longest matching prefix.

21. The computer program according to claim 17, wherein the computer readable medium further performs the steps of inserting a new prefix, wherein the intervals that are created by the new prefix are stored in the basic interval tree; wherein the new prefix is stored in a prefix tree; wherein the pointer to the prefix tree for the new intervals is established and stored with the new intervals in the basic interval tree.

22. The computer program according to claim 17, wherein the computer readable medium further performs the steps of deleting a prefix, wherein the intervals in the basic interval tree that correspond to the prefix to be deleted are deleted; wherein the prefix is deleted from the prefix tree; wherein the pointer to the prefix tree for the deleted intervals are deleted from the basic interval tree node.